#4

Attorney Docket No. 06944.0037-01
Customer Number 22,852

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	)	
	)	
Robert J. LAMBERT et al.	)	Group Art Unit: 2121
	)	
Serial No.: 09/931,013	)	Examiner:
	)	
Filed: August 17, 2001	)	
	)	
For: METHOD FOR ACCELERATING CRYPTOGRAPHIC OPERATIONS ON ELLIPTIC CURVES	)	

Assistant Commissioner for Patents
Washington, DC 20231

Sir:

CLAIM FOR PRIORITY

Under the provisions of 35 U.S.C. § 119, Applicants hereby claim the benefit of the filing date of Canadian Patent Application No. 2,257,008 filed December 24, 1998, for the above-identified U.S. patent application.

In support of this claim for priority, enclosed is one certified copy of the priority application.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

Dated: November 30, 2001

By: _____
Ernest F. Chapman
Reg. No. 25,961

EFC/FPD/bl
Enclosures

FINNEGAN
HENDERSON
FARABOW
GARRETT &
DUNNER LLP

1300 I Street, NW
Washington, DC 20005
202.408.4000
Fax 202.408.4400
www.finnegan.com

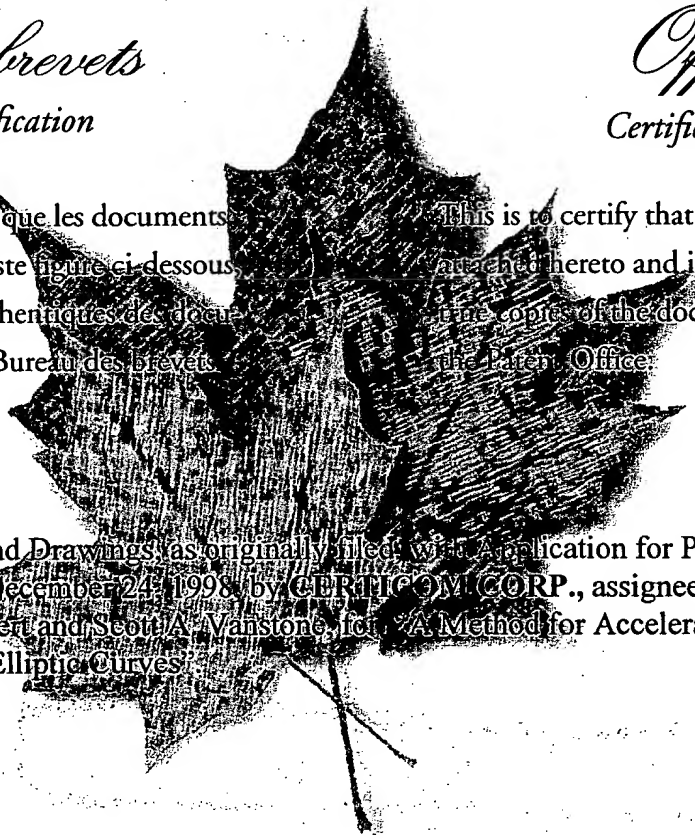
Office de la propriété intellectuelle du Canada
Un organisme d'Industrie Canada

Canadian Intellectual Property Office
An Agency of Industry Canada

OIPE JC177
NOV 30 2001
PATENT & TRADEMARK OFFICE

Bureau canadien des brevets
Certification

Canadian Patent Office
Certification



La présente atteste que les documents ci-joints, dont la liste figure ci-dessous, sont des copies authentiques des documents déposés au Bureau des brevets

This is to certify that the documents attached hereto and identified below are true copies of the documents on file in the Patent Office

Specification and Drawings, as originally filed, with Application for Patent Serial No: **2,257,008**, on December 24, 1998, by **CERTICOM CORP.**, assignee of Robert Gallant, Robert J. Lambert and Scott A. Vanstone, for "A Method for Accelerating Cryptographic Operations on Elliptic Curves"

CERTIFIED COPY OF PRIORITY DOCUMENT

S. Grégoire
Agent certificateur/Certifying Officer

September 19, 2001
Date

Canada

(CIPO 68)
01-12-00

OPIC CIPO

ABSTRACT

This invention provides a method for accelerating multiplication of an elliptic curve point Q(x,y) by a scalar k, the method comprising the steps of selecting an elliptic curve over a finite field Fq where q is a prime power such that there exists an endomorphism ψ, where $\psi(Q) = \lambda \cdot Q$ for all points Q(x,y) on the elliptic curve; and using smaller representations ki of the scalar k in combination with the mapping ψ to compute the scalar multiple of the elliptic curve point Q.

A METHOD FOR ACCELERATING CRYPTOGRAPHIC OPERATIONS ON ELLIPTIC CURVES

This invention relates to a method for accelerating certain cryptographic operations on a class of elliptic curves.

BACKGROUND OF THE INVENTION

Public-key data communication systems are used to transfer information between a pair of correspondents. At least part of the information exchanged is enciphered by a predetermined mathematical operation by the sender and the recipient may perform a complementary mathematical operation to decipher the information.

Various protocols exist for implementing such a scheme and some have been widely used. In each case however the sender is required to perform a computation to sign the information to be transferred and the receiver is required to perform a computation to verify the signed information.

In a typical implementation a signature component s has the form:-

$s = ae + k \pmod n$

where:

P is a point on the curve which is a predefined parameter of the system;

k is a random integer selected as a short term *private* or session key;

R = kP is the corresponding short term *public* key;

a is the long term private key of the sender;

Q = aP is the senders corresponding public key;

e is a secure hash, such as the SHA-1 hash function, of a message *m* and the short term public key R; and

n is the order of the curve.

The sender sends to the recipient a message including m, s, and R and the signature is verified by computing the value $R' = (sP-eQ)$ which should correspond to R. If the computed values correspond then the signature is verified.

In order to perform the verification it is necessary to compute a number of point multiplications to obtain sP and eQ, each of which is computationally complex. Where the

recipient has adequate computing, power this does not present a particular problem but where the recipient has limited computing power, such as in a secure token or a "Smart card " application, the computations may introduce delays in the verification process.

Elliptic curve cryptography (ECC) provides a solution to the computation issue. ECC permits reductions in key and certificate size that translate to smaller memory requirements, which represent significant cost savings. ECC can not only significantly reduce the cost, but also accelerate the deployment of smart cards in next-generation applications. Additionally, although the ECC algorithm allows for a reduction in key size, the same level of security as other algorithms with larger keys is maintained.

However, there is still a need to perform faster calculations on the keys so as to speed up the information transfer while maintaining a low cost of production of data transfer circuits.

It is therefore an object of the present invention to provide a method and apparatus in which at least some of the above disadvantages are obviated or mitigated.

SUMMARY OF THE INVENTION

In general terms, the present invention provides an opportunity to reduce the complexity required to perform an integral part of the cryptographic process, thereby increasing the speed at which it may be performed and reducing the computing power required to perform it.

The present invention is a method for speeding up certain cryptographic operations on elliptic curves. The method is based on the following observation that given an elliptic curve having complex multiplication over a finite field. Then there is an λ, which is the solution to a quadratic, for which a complex multiplication mapping is equivalent to multiplying a point by λ. It will often be less computationally expensive to compute via λQ the complex multiplication map, compared to treating λ as a integer and performing the EC multiplication. In practice, point multiplication by other scalars (not just λ) is required. The multiplication by λ map may be used to compute other multiples.

In accordance with this invention there is provided a method for accelerating multiplication of an elliptic curve point Q(x,y) by a scalar k, the method comprising the steps of:

selecting an elliptic curve over a finite field Fq where q is a prime power such that there exists an endomorphism ψ, where $\psi(Q) = \lambda \cdot Q$ for all points Q(x,y) on the elliptic curve; and using smaller representations k_i of the scalar k in combination with the mapping ψ to compute the scalar multiple of the elliptic curve point Q.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other features of the preferred embodiments of the invention will become more apparent in the following detailed description in which reference is made to the appended drawings wherein:

Figure 1 is a schematic diagram of a communication system;

Figure 2 is a flow chart showing a general method according to an embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

For convenience in the following description, like numerals refer to like structures in the drawings. Referring to Figure 1, a data communication system 10 includes a pair of correspondents, designated as a sender12, and a recipient 14, who are connected by a communication channel 16. Each of the correspondents 12,14 includes a cryptographic processor 18,20 respectively that may process digital information and prepare it for transmission through the channel 16 as will be described below. Each of the correspondents 12,14 also includes a computational unit 19,21 respectively to perform mathematical computations related to the cryptographic processors 18,20. The computational power of the units 19,21 will vary according to the nature of the correspondents 12,14 but for the purpose of the present disclosure, it will be assumed that the unit 19 has greater power than that of unit 21, which may in fact be a Smart card or the like. Cryptographic computations such as the multiplication of an elliptic curve point by a scalar value are computationally expensive.

One of the functions of the cryptographic processor 18 is to perform point multiplications of the form $k \cdot Q$ so that it may be used in a cryptographic scheme. In accordance with this invention there is provided a method for accelerating scalar multiplication of an elliptic curve point Q(x,y) is shown generally by the numeral 50. The subject algorithm

increases the speed at which the processors 12 can for example sign and verify messages for specific classes of elliptic curves. The method based on the observation that given the general equation for an elliptic curve E:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \quad (1)$$

over a finite field F_q (q is a prime power) and when there exists an endomorphism ψ, where $\psi(Q) = \lambda \cdot Q$ for all points Q(x,y) on the elliptic curve, then multiplication of the point Q by an integer k may be accelerated by utilizing combinations of smaller representations k_i of k in combination with the mapping ψ. The mapping ψ also allows precomputation of group elements and combinations thereof, which may be used in subsequent calculation of kQ.

Referring now to figure 2, a flow chart of a general embodiment for accelerating point multiplication on an elliptic curve, according to the present invention, is shown by numeral 50. The system parameters are first determined. This involves selecting an elliptic curve. In a first embodiment of the invention the generalized elliptic curve (1) may be expressed in the following form:

$$E: y^2 = x^3 + b \bmod p; \text{ where p is a prime} \quad (2)$$

Firstly, the modulus p can be determined such that there is a number, γ where $\gamma \in F_p$ (F_p is the field of size p consisting of all 'integers mod p'), and $\gamma^3 \equiv 1 \bmod p$ (a cube root of unity). If for example p = 7, then $\gamma = 2$, since $2^3 \bmod 7 = 1$. Such a γ does not necessarily exist for all p, and therefore it must be taken into consideration when choosing the value of p. Typically, the chosen p should be at least 160 bits in length for adequate cryptographic strength.

Consider next the mapping function ψ: $(x, y) \rightarrow (\gamma x, y)$, which simply maps one set of points on the curve to another set of points on the curve. Then, there exists an integer λ such that $\psi(Q) = \lambda \cdot Q$ for all points Q(x,y) on the elliptic curve, E. This integer λ may be found by noting that $\lambda^3 \equiv 1 \bmod n$, where n is the number of points on the elliptic curve E over F_p i.e. no. of points on $E(F_p)$. There may exist more than one solution for λ in $\lambda^3 \equiv 1 \bmod n$, but only one of those solutions will satisfy the mapping function ψ. It is important to note that since $\gamma^3 \bmod p = 1$, both Q and $\psi(Q)$ satisfy the equation for E. Therefore, instead of having to perform lengthy calculations to determine the results of multiplication by λ, it can be done very

efficiently using the results of the mapping function. That is, multiplication by λ can be done very efficiently.

These system parameters E, p, Q, λ, $\psi(Q)$, and γ may now be stored in the card 12 at manufacture time for use by the cryptographic processor 18. The number of k_i's being used may also be determined at this time.

It is possible to select a value for k such that:

$$k = (k_0 + k_1\lambda) \bmod n \qquad (3)$$

where n is the number of points on $E(F_p)$. For cryptographic operations such as encryption and Diffie-Hellman, the processor 12 randomly generates the value for k and then computes $k \cdot Q$ (where Q is a point on E). In these cases, it would be possible to select values for k_0 and k_1 at random, having a length of $[\log_2 (n)]/2$ not including sign bits, and then calculate the value for k using equation (3). (i.e. the length of the k_i's are chosen to be at least one half the length k). The point $k \cdot Q$ then becomes:

$$k \cdot Q = (k_0 Q + k_1 \lambda Q) \bmod n \qquad (4)$$

Now, the right side of equation (4) can be calculated quickly using an algorithm analogous to the "Simultaneous Multiple Exponentiation" as described in the "Handbook of Applied Cryptography" by Menezes et. al.(Algorithm 14.88). For convenience the algorithm is reproduced below. It may be noted that in an additive group exponentiation is analogous to addition, thus replacing the multiplication in the algorithm with addition, yields the following:

Algorithm 1 Simultaneous Multiple Addition

INPUT: group elements $g_0, g_1, \ldots, g_{l-1}$ and non negative t-bit integers $e_0, e_1, \ldots, e_{l-1}$.
OUTPUT: $g_0e_0 + g_1e_1 + \ldots + g_{l-1}e_{l-1}$.

step1. *Precomputation.* For i from 0 to $(2^l - 1)$:

$$G_i \leftarrow \Sigma_{j=0}^{l-1} g_j i_j$$

where $i = (i_{l-1} \ldots i_0)_2$

step2. $A \leftarrow 0$

step3. For i from 1 to t do the following:

$$A \leftarrow A + A, A \leftarrow A + G_{I_i}$$

step4. Return (A) where $A = g_0e_0 + g_1e_1 + \ldots + g_{l-1}e_{l-1}$

Applying this algorithm to equation (4) it can be seen that there are two group elements, Q and λQ, and therefore $l = 2$. The results of Precomputation with $l = 2$ is shown in table 1.

i	0	1	2	3
G_i	0	g_0	g_1	$g_0 + g_1$

Table 1.

In the present situation, k_0 through k_1 is analogous to e_0 through e_1, and g_0 through g_1 is analogous to Q and $\psi(Q)$ respectively. It is straightforward to compute $\psi(Q) = \lambda \cdot Q = (\gamma x, y)$. The next step is to construct the following point: $Q + \psi(Q)$. Thusly, it is possible to fill in table 1 with the computed elements to yield table 2. These elements may be pre-computed and stored in memory.

i	0	1	2	3
G_i	0	Q	$\psi(Q)$	$Q + \psi(Q)$

Table 2.

Next a notional matrix or combing table may be constructed using the binary representation of k_i. If, for example, $k_0 = 30$ and $k_1 = 10$, then the notional matrix constructed from their binary representation is shown in Table 3.

	I_1	I_2	I_3	I_4	I_5
K_0	1	1	1	1	0
K_1	0	1	0	1	0

Table 3

Before step3 of the algorithm can be performed, I_1 through I_t have to be found. Here t has the value five since the maximum number of bits in the binary representation of k_0 through k_1 is five. I_i is determined by the number represented in the i^{th} column where the first row contains

the least significant bit. Therefore it can be seen from table 3 that $I_1 = 1$, $I_2 = 3$, $I_3 = 1$, $I_4 = 3$, and $I_5 = 0$. All the components needed to complete the algorithm 64 are available and the results of step three are shown in table 4.

i	A
1	Q
2	$3Q + \psi(Q)$
3	$7Q + 2\psi(Q)$
4	$15Q + 5\psi(Q)$
5	$30Q + 10\psi(Q)$

Table 4

Thus it may be seen that this method will require a number of point doubles equal to $\max\{\log_2(k_i)\}$, and almost as many point additions. The number of point additions can be reduced using windowing (Alg. 14.85 HAC) and exponent recoding techniques. The point additions are easily performed by retrieving the appropriate precomputed element G_i from table 2. To summarize, for cryptographic operations like encryption and Diffie-Hellman, where one must pick an integer k and compute points $k \cdot Q$. One can first choose k_0 and k_1 at random, each having a length one half the length of n. When the k's are chosen in this way, the method seems to be as secure as the normal methods. Of course it is possible to choose the k_i's to have fewer bits set in order to trade off between efficiency and security.

In a second embodiment of the invention a different form of the generalized elliptic curve equation (1) is used, as show below.

$$y^2 = (x^3 - ax) \bmod p \qquad (5)$$

Once again, p will be a prime number having at least 160 bits. For this type of curve, the properties required for γ are different. It is now required to find a value such that $\gamma^2 = -1 \bmod p$. A change in the property of γ requires a different mapping function ψ' to be used. In this embodiment the mapping takes the form ψ': $(x, y) \rightarrow (-x, \gamma y)$. If (x,y) is on the curve, then $\psi'(x,y)$ is on the curve. It is possible to note that $\lambda^4 \equiv 1 \bmod n$ (n is still the number

of point on $E(F_p)$), and therefore λ can be calculated. The mapping $\psi'(Q) = \lambda \cdot Q$ as before and once again multiplication by λ can be done very efficiently for this curve. The equation for k in this embodiment is the same as in the first embodiment and is represented by:

$$k = (k_0 + k_1\lambda) \bmod n \qquad (6)$$

This equation is the same as in the second embodiment, having only two group elements. Thus using the group elements Q and $Q + \psi'(Q)$ in the algorithm 1, the point $k \cdot Q$ may be calculated. This computation will require a number of point doubles equal to $\max\{\log_2(k_i)\}$, and a similar number of point additions. Thus described earlier the number of point additions can be reduced using windowing and exponent recoding techniques.

This method applies to other elliptic curves, so long as there exists an efficiently computable endomorphism, ψ. For cryptographic protocols, where we do not get to choose k, we must first find k_0, k_1 of the desired "short" form such that $k = (k_0 + k_1\lambda) \bmod n$. This could be done using the L^3 algorithm, for example.

As may be seen in the embodiments described above when a point is known beforehand, tables can be built to speed multiplication. However, there are cases when multiples of previously unknown points are required (for example, this can occur in ECDSA verification). That is a k is provided and it is necessary then to determine suitable representations for k_i.

Thus in a third embodiment according to the present invention, the point Q, the required multiple k, and the complex multiplication multiple λ are known. It is necessary to determine the k_i's since the value for k is predetermined. A method for doing this described as follows. As a pre-computation (not requiring k) we compute two relations:

$$a_0 + b_0\lambda \equiv 0 \bmod n$$

$$a_1 + b_1\lambda \equiv 0 \bmod n$$

such that a_i and b_i are numbers smaller than n. It is preferable that a_i and b_i are as small as possible, however, the present method has advantages even when a_i and b_i are not minimal. Typically the method produces k_0 and k_1 having representations one half the size of the original k.

To produce small a_i and b_i, it is possible to make use of the LLL algorithm, but in this preferred embodiment the simple extended Euclidean algorithm is employed on the pair (n, λ).

The extended Euclidean algorithm on (n, λ) produces linear combinations $c_i n + d_i \lambda = r_i$, where the representation of r_i (e.g. bit-length) decreases and the representation of c_i and d_i increases with i.

The two smallest values of $|(d_i, r_i)|$ resulting from using the extended Euclidean algorithm are saved. The size of these vectors are measured with the squared Euclidean norm $|(d_i, r_i)| = d_i^2 + r_i^2$. Denote the terms in these minimal relations $\hat{d}_0, \hat{r}_0$ and $\hat{d}_1, \hat{r}_1$. Such relations will typically occur in the middle of the algorithm. Even if the minimal relations are not retained, suboptimal relations may still give the method an advantage in the calculation of point multiples.

It is possible to construct a_i and b_i by setting $a_0 = -\hat{r}_0$, $b_0 = \hat{d}_0$ and $a_1 = -\hat{r}_1$, $b_1 = \hat{d}_0$. The next task is to find a small representation for the multiple k.

The relations can be viewed as vectors $\mathbf{u_o} = (a_0, b_0)$ and $\mathbf{u_1} = (a_1, b_1)$. These vectors satisfy $a_i + b_i\lambda = 0 \pmod n$. Define multiplication of the group elements Q by the vector $\mathbf{v} = (v_0, v_1)$ as $(v_0 + v_1\lambda)Q$. Since $a_i + b_i\lambda = 0 \pmod n$, we have $\mathbf{u_o}R = \mathbf{u_1}R = 0$ for any group element R. Hence for any integers z_0 and z_1 we have $\mathbf{v}R = (\mathbf{v} - z_0\mathbf{u_0} - z_1\mathbf{u_1})R$ for any group element R.

Integers z_0 and z_1 may be chosen such that the vector $\mathbf{v}' = \mathbf{v} - z_0\mathbf{u_0} - z_1\mathbf{u_1}$ has components that are as small as possible.

Again, this method will have an advantage if $\mathbf{v}'$ is small, but not necessarily minimally so. The appropriate z_0 and z_1 are calculated by converting the basis of v into the basis $\{\mathbf{u_0}, \mathbf{u_1}\}$. The conversion between basis involves matrix multiplication. To convert the vector $\mathbf{v} = (v_0, v_1)$ from the $\{\mathbf{u_0}, \mathbf{u_1}\}$ basis to the standard orthonormal basis $\{(1,0),(0,1)\}$,

$$v_{\{(1,0),(0,1)\}} = v_{(u_0,u_1)}M = (v_0, v_1)\begin{bmatrix} a_0 & b_0 \\ a_1 & b_1 \end{bmatrix}$$

To convert in the other direction, from the standard orthonormal basis $\{(1,0),(0,1)\}$ to the $(\mathbf{u_0}, \mathbf{u_1})$ basis, the multiplication is simply by the inverse of M,

$$v_{(u_0,u_1)} = v_{\{(1,0),(0,1)\}}inverse(M) = v_{\{(1,0),(0,1)\}}\frac{1}{a_0b_1 - a_1b_0}\begin{bmatrix} b_1 & -b_0 \\ -a_1 & a_0 \end{bmatrix}$$

Since the vector $\mathbf{v} = (k, 0)$ has a zero component, the bottom row of inverse(M) is not required, and therefore to convert to the $\{\mathbf{u}_0, \mathbf{u}_1\}$ basis only the fractions

$$f_0 = \frac{b_1}{a_0 b_1 - a_1 b_0}$$

and

$$f_1 = \frac{b_0}{a_0 b_1 - a_1 b_0}$$

are needed.

Calculate $\mathbf{z} = (z_0, z_1)$, where z is defined as $(z_0, z_1) = (\text{round}(kf_0), \text{round}(kf_1))$. The fractions f_0 and f_1 may be precomputed to enough precision so that this operation may be effected only with multiplication. Note also that the computations leading to these fractions do not depend upon k, therefore they can be computed once when the elliptic curve is chosen as a system parameter, and do not need to be recalculated for each k.

Other vectors near to **z** will also be useful, therefore could be replaced with floor or ceiling functions or some other approximation.

Now that a suitable **z** has been determined, an efficient equivalent to **v** is calculated by $\mathbf{v}' = (v_0', v_1') = \mathbf{v} - z_0\mathbf{u}_0 - z_1\mathbf{u}_1$. The phrase "efficient equivalent" implies a vector **v'** such that $\mathbf{v}'P = \mathbf{v}P$ and **v'** has small coefficients. The value kQ is then calculated as $v_0'Q + v_1'\lambda Q$. This value can be calculated using simultaneous point addition, and non-adjacent form recoding can also have advantages.

Using these methods to determine the value of $k \cdot Q$ greatly reduces the processing power required by the cryptographic processors 12. It also increases the speed at which these repetitive calculations can be done which, in turn, reduces the time to transfer information.

Although the invention has been described with reference to certain specific embodiments, various modifications thereof will be apparent to those skilled in the art without departing from the spirit and scope of the invention as outlined in the claims appended hereto.

THE EMBODIMENTS OF THE INVENTION IN WHICH AN EXCLUSIVE PROPERTY OR PRIVILEGE IS CLAIMED ARE DEFINED AS FOLLOWS:

1. A method for accelerating multiplication of an elliptic curve point Q(x,y) by a scalar k, the method comprising the steps of:

a) selecting an elliptic curve over a finite field Fq where q is a prime power such that there exists an endomorphism ψ, where $\psi(Q) = \lambda \cdot Q$ for all points Q(x,y) on the elliptic curve; and

b) using smaller representations ki of the scalar k in combination with the mapping ψ to compute the scalar multiple of the elliptic curve point Q.

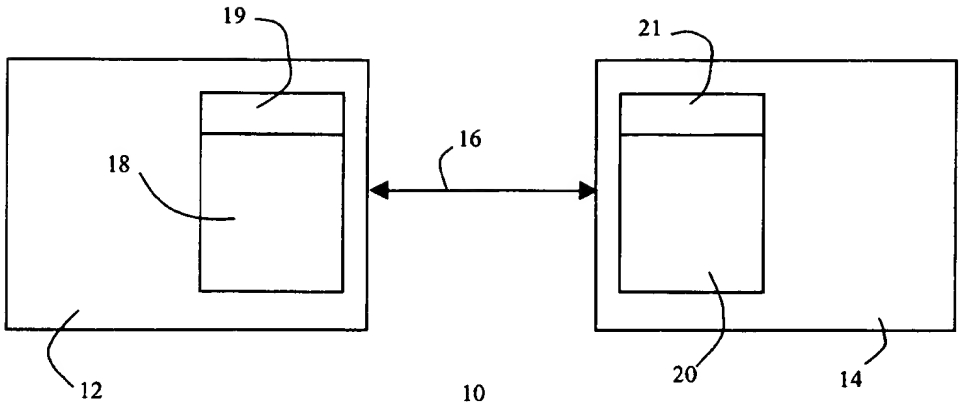


Figure 1

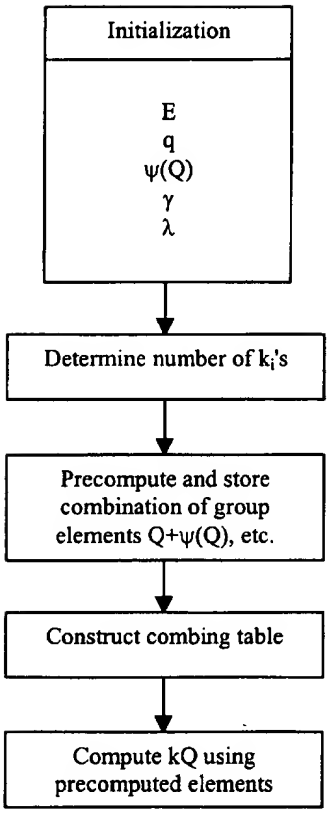

Initialization
E
q
ψ(Q)
γ
λ
Determine number of k_i's
Precompute and store combination of group elements Q+ψ(Q), etc.
Construct combing table
Compute kQ using precomputed elements

Figure 2

THIS PAGE BLANK (USPTO)